SRI International

LOCAL PRAGMATICS

Technical Note 429

December 11, 1987

By: Jerry R. Hobbs, Sr. Computer Scientist
and
Paul Martin, Computer Scientist

Artificial Intelligence Center
Computer and Information Sciences Division

**APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED**

This research was funded by the Defense Advanced Research Projects Agency under the Office of Naval Research contract N00014-85-C-0013.

SRI International

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Report Documentation Page	*Form Approved OMB No. 0704-0188*

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
11 DEC 1987		**00-12-1987 to 00-12-1987**

4. TITLE AND SUBTITLE	
Local Pragmatics	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SRI International,333 Ravenswood Avenue,Menlo Park,CA,94025**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**30**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Local Pragmatics

Jerry R. Hobbs and Paul Martin
Artificial Intelligence Center
SRI International

Abstract

The outline of a unified theory of local pragmatics phenomena is presented, including an approach to the problems of reference resolution, metonymy, and interpreting nominal compounds. The TACITUS computer system embodying this theory is also described. The theory and system are based on the use of a theorem prover to draw the appropriate inferences from a large knowledge base of commonsense and technical knowledge. Issues of control are discussed. Two important kinds of implicatures are defined, and it is shown how they can be used to determine what in a text is given and what is new.

1 The Problems

In the messages about breakdowns in machinery that are being processed by the TACITUS system at SRI International, we find the following sentence:

(1) We disengaged the compressor after the lube oil alarm.

This sentence, like virtually every sentence in natural language discourse, confronts us with difficult problems of interpretation. First, there are the reference problems; what do "the compressor" and "the lube oil alarm" refer to. Then there is the problem of interpreting the implicit relation between the two nouns "lube oil" (considered as a multiword) and "alarm" in the nominal compound "lube oil alarm". There is also a metonymy that needs to be expanded. An alarm is a physical object, but "after" requires events for its arguments. We need to coerce "the lube oil alarm" into "the sounding of the lube oil alarm".[1] There is the syntactic ambiguity problem

[1] One could say that "alarm" in this sentence means the event of "alarming", so that there is no metonymy. If we took this approach, however, there would be a lexical ambi-

of whether to attach the prepositional phrase "after the lube oil alarm" to "the compressor" or to "disengaged".

All of these problems we have come to call problems in "local pragmatics". Local pragmatics encompasses reference resolution, metonymy, the interpretation of nominal compounds and other implicit and vague predicates, and the resolution of syntactic, lexical, and quantifier scope ambiguities. It may be that to solve these problems, we need to look at the surrounding discourse and the context in which the utterance is made. But we can determine locally—just from the sentence itself—that we *have* a problem. They seem to be specifically linguistic problems, but the traditional linguistic methods in syntax and semantics have not yielded solutions of any generality.

The difficulty, as is well-known, is that to solve these problems we need to use a great deal of arbitrarily detailed general commonsense and domain-specific technical knowledge. In sentence (1) we need to know, for example, that the compressor has a lube oil system, which has an alarm, which sounds when the pressure of the lube oil drops too low. We need to know that disengaging and sounding are events, and that a compressor isn't.

A theory of local pragmatics phenomena must therefore be a theory about how knowledge is used. The aim of our research has been to develop a unified theory of local pragmatics, based on the drawing of appropriate inferences from a large knowledge base, and to implement a system embodying that theory for solving local pragmatics problems in naturally occurring texts. It is our intention that in this theory general solutions to local pragmatics problems can be characterized, but it should also be possible to cast current, limited approaches to these phenomena as special cases of the general solutions.

This research is taking place in the context of the TACITUS project,[2] the specific aim of which is to develop interpretation processes for handling casualty reports (casreps), which are messages in free-flowing text about breakdowns in mechanical devices. More broadly, however, its aim is to develop general procedures, together with the underlying theory, for using commonsense and technical knowledge in the interpretation of written (and spoken) discourse regardless of domain. We expect such interpretation processes to constitute an essential component, and indeed the principal

guity problem of deciding which sense of "alarm" is being used, and the processing saved on metonymy would be used up by the correspondingly more difficult nominal compound problem.

[2]A part of the Strategic Computing program sponsored by the Defense Advanced Research Projects Agency.

component, in sophisticated natural language systems of the future.

The TACITUS system has four principal components. First, a syntactic front-end, the DIALOGIC system (Grosz et al., 1982), translates sentences of a text into a logical form in first-order predicate calculus, described in Section 3.1. Second, we are building a knowledge base, specifying large portions of potentially relevant knowledge encoded as predicate calculus axioms (Hobbs et al., 1986). Third, the TACITUS system makes use of the KADS theorem prover, developed by Mark Stickel (Stickel, 1982). Finally, there is the pragmatics component, which uses the theorem prover to draw appropriate inferences from the knowledge base, thereby constructing an interpretation of the text. At the present time, the pragmatics component deals only with local pragmatics, and what it does is the subject of this paper. In addition, however, we are beginning to augment the pragmatics component with procedures for relating the text to the user's interests, and we plan to augment it with procedures for recognizing discourse structure.

Section 2 describes the three local pragmatics problems we are currently devoting our efforts to. The solutions to each of them requires constructing and proving a particular logical expression. In Section 3 we discuss how an expression—the interpretation expression—is constructed for an entire sentence, such that its proof constitutes an interpretation of the sentence. We also discuss how the search for a proof of this expression can be ordered. Very often, interpretation requires that certain facts be assumed, where the only warrant for the assumptions is that they lead to a good interpretation. These are called "implicatures". In Section 4 we describe our current approach to implicature and an approach we are just beginning to investigate. In Section 5 we describe and illustrate the current implementation.

2 Local Pragmatics Phenomena

2.1 Interpretation as Deduction

Language does not give us meanings. Rather, it gives us problems to be solved by reasoning about the sentence, using general knowledge. We get meaning only by solving these problems. Before we can use what is asserted in a sentence to draw further conclusions, we must first interpret the sentence by deducing its presuppositions from the knowledge base.

Since knowledge is encoded in the TACITUS system as axioms in predicate calculus, reasoning about them, and hence arriving at interpretations, is a matter of deduction. To interpret a sentence, we first determine from the

sentence what interpretation problems we are required to solve, i.e., what local pragmatics phenomena are exhibited. These are framed as expressions to be proved by the deduction component. The proofs of these expressions constitute the interpretation of the sentence. Where there is more than one interpretation, it is because there is more than one proof for the expressions.

In this section, we describe the three phenomena we are addressing first—reference, metonymy, and nominal compounds. For each of these, we describe the expression that needs to be proved. For the last two, we describe how current standard techniques can be seen as special cases of our general approach.

2.2 Reference

Entities are referred to in discourse in many guises. They can appear as proper nouns, definite, indefinite, and bare noun phrases of varying specificity, pronouns, and omitted or implicit arguments. Moreover, verbs, adverbs, and adjectives can refer to events, conditions, or situations. The problem in all of these cases is to determine what is being referred to. Here we confine ourselves to definite noun phrases, although in Section 4 we extend our treatment to indefinite and bare noun phrases and nonnominal reference.

In the sentence

The alarm sounded.

the noun phrase "the alarm" is definite, and the hearer is therefore expected to be able to identify a unique entity that the speaker intends to refer to. Restating this in theorem-proving terminology, the natural language system should be able to prove constructively the expression

$$(\exists\, x) alarm(x)$$

That is, it must find an x which is an alarm in the model of the domain. If it succeeds, it has solved the reference problem.[3]

Similarly, in the text

(2) The compressor is down.
The air inlet valve is clogged.

[3] In this paper we ignore the problem of the uniqueness of the entity referred to. A hint of our approach is this: If the search for a proof is heuristically ordered by salience, then the entity found will be the uniquely most salient.

we need, in interpreting the second sentence, to prove the existence of an air inlet valve. We know from the first sentence that there is a compressor, and our model of the domain tells us that compressors have air inlet valves. So we can conclude that the reference is to the air inlet valve of that compressor.

In processing the casreps there is a further wrinkle in the problem—noun phrases rarely have determiners, and there is no clear signal whether it is definite or indefinite. This problem is dealt with in Section 4.

2.3 Metonymy

In metonymy, or indirect reference, we refer to one thing as a way of referring to something related to it. Sentence (1) contains the phrase "after the alarm", where what is really meant is "after *the sounding of* the alarm". "The alarm" is used to refer to the sounding which is related to it, and in interpreting the phrase we need to *coerce* the alarm to its sounding.

Metonymy is extremely common in discourse; when examined closely, very few sentences will be found without an example. Certain functions very frequently provide the required coercions. Wholes are used for parts; tokens are used for types; people are used for names. Nunberg (1978), however, has shown that there is no finite set of possible coercion functions. The relation between the explicit and implicit referents can be virtually anything.

From a generation point of view, the story behind metonymy must go something like this: A speaker decides to say

$$\ldots \wedge after(E_0, E_1) \wedge sound'(E_1, A) \wedge alarm(A)$$

that is, E_0 is after the sounding E_1 of the alarm A. However, given the first and last predications, the middle one is obvious, and hence can be left out. Since $after$ needs a second argument and A has to be the argument of something, $after$ takes A as its second argument, yielding

$$\ldots \wedge after(E_0, A) \wedge alarm(A)$$

or "after the alarm".

From an interpretation point of view, the story is this: Every morpheme in a sentence corresponds to a predication, and every predicate imposes *selectional constraints* on its arguments. Since entities in the text are generally the arguments of more than one predicate, there could well be inconsistent constraints imposed on them (especially in light of the above generation story). To eliminate this inconsistency, we interpose, as a matter of course, another entity and another relation between any two predications. Thus, when we encounter in the logical form of a sentence

$$\ldots \wedge\ after(e_0, a) \wedge alarm(a)$$

we assume that what is intended is really

$$\ldots \wedge\ after(e_0, k) \wedge rel(k, a) \wedge alarm(a)$$

for some entity k and some relation rel. The predication $rel(k, a)$ functions as a kind of buffer, or impedence match, between the explicit predications with their possibly inconsistent constraints. In many cases, of course, there is no inconsistency. The argument satisfies the selectional constraints imposed by the predicate. In these cases, k is a and rel is identity. This in fact is the first possibility tried in the implemented system. Where this fails, however, the problem is to find what k and rel refer to, subject to the constraint, imposed by the predicate $after$, that k is an event.

Therefore, TACITUS modifies the logical form of the sentence to

$$\ldots \wedge\ after(e_0, k) \wedge rel(k, a) \wedge alarm(a)$$

and for an interpretation, the expression that must be proved constructively is

$$(\exists\, k, rel, a) event(k) \wedge rel(k, a) \wedge alarm(a)$$

We need to find an event k bearing some relation rel to the alarm.

The most common current method for dealing with metonymy, e.g., in the TEAM system (Grosz et al., 1985), is to specify a small set of possible coercion functions, such as *name-of*. This method can be captured in the present framework by treating rel not as a predicate variable, but as a predicate constant, and expressing the possible coercions in axioms like the following:

$$(\forall\, x, y) name(x, y) \supset rel(x, y)$$

That is, if x is the name of y, then y can be coerced to x. This in fact is the method we have implemented in our initial version of the TACITUS system.

2.4 Nominal Compounds

To interpret a nominal compound, like "lube oil alarm" (where "lube oil" is taken as a multiword), it is necessay to discover the implicit relation between the two nouns.[4] Some relations occur quite frequently in nominal

[4]Some nominal compounds can of course be treated as single lexical items. This case is not interesting and is not considered here.

compounds—*part-of*, *location*, *purpose*. Moreover, when the head noun is relational, the modifier noun is often one of the arguments of the relation. Levi (1978) argued that these two cases encompassed virtually all nominal compounds. However, Downing (1977) and others have shown that virtually any relation can occur. A lube oil alarm, for example, is an alarm that sounds when the pressure of the lube oil drops too low.

To discover the implicit relation, one must prove constructively from the knowledge base the existence of some possible relation, which we may call nn, between the entities referred to by the nouns:

$$(\exists\, x, y) alarm(x) \wedge \textit{lube-oil}(y) \wedge nn(y, x)$$

Just as with metonymy, the most common method for dealing with nominal compounds[5] is to hypothesize a small set of possible relations, such as *part-of*. In our framework, we can use this approach by taking nn to be not a predicate variable but a predicate constant, and encoding the possibilities in axioms like

$$(\forall\, x, y) part(x, y) \supset nn(y, x)$$

For example, if a blade x is a part of a fan y, then "fan blade" is a possible nominal compound. Equality also implies an nn relation, for nominal compounds like "metal particle" (an x such that x is metal and x is a particle).

To deal with relational nouns, such as "oil sample" and "oil pressure", we encode axioms like

(3) $(\forall\, x, y) sample(x, y) \supset nn(y, x)$

This tells us that if x is a sample of oil y, then x can be referred to by the nominal compound "oil sample".

Finin (1980) argues that one of the most common kinds of relations is one that involves the function of the referent of the head noun. The function of a pump is to pump a fluid, so "oil pump" is a possible nominal compound. This can be encoded in axioms of the pattern

$$(\forall\, x, y, e) function(e, x) \wedge p'(e, x, y) \supset nn(y, x)$$

That is, if e is the function of x where e is the situation of x doing something p to y, then there is an nn relation between y and x.

As with metonymy, in our initial version of TACITUS, it is the standard, restricted method that we have implemented. This is because we wanted

[5] Other than treating them as multiwords.

to make sure we were not losing ground in seeking a general solution. Nevertheless, our approach allows us to begin experimenting with the general solution to the nominal compound problem, where the implicit relation can be anything at all.

3 The Construction and Proof of the Interpretation Expression

3.1 Preliminary Note on Logical Form

DIALOGIC, the syntactic front end of TACITUS, produces a logical form for the sentence in something like a first-order logic but encoding grammatical subordination relations as well as predicate-argument relations. It is "ontologically promiscuous" in that events and conditions are reified (Hobbs, 1985a). A slightly simplified version of the logical form for the sentence

(4) The lube oil alarm sounded.

is

(5) $past([e_1 \mid sound'(e_1, [a_1 \mid alarm(a_1) \wedge nn([o_1 \mid lube\text{-}oil(o_1)], a_1)])])$

"$\mid$" can be read "such that" or "where", so that a paraphrase of this formula would be "In the past there was an event e_1 which was a sounding event by a_1 where a_1 is an alarm and there is an nn relation between a_1 and o_1 such that o_1 is lube oil.

In general, the logical form of a sentence is a "proposition". A proposition is a predicate applied to one or more arguments. An argument is either a variable or a "complex term". A complex term is a variable, followed by a "such that" sign, followed by a "restriction". (Complex terms are surrounded by square brackets for readability.) A restriction is a conjunction of propositions.

This notation can be translated into a notation using four-part quantifier structures (Woods, 1977; Moore, 1981) by successively applying the following transformation:

$$p([x \mid q(x)]) \Rightarrow (\exists x \ q(x) \ p(x))^6$$

[6]Quantifiers other than existentials are ignored in this paper. For the treatment we intend to give them, see Hobbs (1983).

It can be translated into standard Russellian notation, with a consequent loss of information about grammatical subordination, by successively applying the following transformation:

$$p([x \mid q(x)]) \Rightarrow p(x) \wedge q(x)$$

3.2 Order of Interpretation

As we saw in Section 2, interpretation involves solving a number of problems, or proving a number of expressions, and this raises a question. In which order should we try to solve them? A naive answer would be to try to solve them "from the inside out". Before trying to find the lube oil *alarm*, we should try to find the lube oil the alarm is an alarm *for*. Before checking that the lube oil alarm obeys the selectional constraints imposed by "sound", we should learn as much as we can about the lube oil alarm; in particular, we should resolve the reference of "the lube oil alarm" so we know what lube oil alarm is being talked about.

This means that given the logical form (5), we should solve the local pragmatics problems in the following order:

1. Find the reference of o_1, the lube oil. Prove

   $$(\exists\, o_1) lube\text{-}oil(o_1)$$

2. Given that, find the reference of a_1, the alarm, and as a by-product, find the implicit relation nn encoded in the nominal compound. If o_1 was resolved to O, then prove

   $$(\exists\, a_1) alarm(a_1) \wedge nn(a_1, O)$$

3. Given that, check the predicate-argument congruence of *sound* applied to a_1. If a_1 was resolved to A and *sound* requires its argument to be a physical object, then prove

   $$(\exists\, k) physical\text{-}object(k) \wedge rel(k, A)$$

Unfortunately, this order will not always work. Information relevant to the solution of any of these local pragmatics problems can come from the solutions of any of the others. For example, in the sentence

This thing won't work.

selectional constraints imposed by "work" provide more information about the referent of "this thing" than the noun phrase itself does.

Thus, in a more sophisticated approach, we would construct a single expression to be proved, encoding what is required for *all* of the local pragmatics problems. For sentence (4), the expression would be

$$(\exists\, k, a_1, nn, o_1) \mathit{physical\text{-}object}(k) \wedge \mathit{rel}(k, a_1) \wedge \mathit{alarm}(a_1) \\ \wedge\, \mathit{nn}(a_1, o_1) \wedge \mathit{lube\text{-}oil}(o_1)$$

Let us call this the *interpretation expression.*

The conjuncts of the interpretation expression could be proved in any order. The inside-out order is only one possibility. The search for a proof is a heuristic, depth-bound, breadth-first search, and the inside-out order can be taken as an indication of how much of its resources the theorem prover should devote to proofs of the various conjuncts, and how early. More resources should be devoted earlier to the initial conjuncts in inside-out order. But other possible orders of proof must be left open. The difficulty with this approach, however, is that it is hard to get partial results in cases of failure.

We are currently using a compromise between these two orders—a fail-soft, inside-out order. As we proceed inside out, at each step the theorem-prover is given the full expression built up to that point. However, the expression has as an antecedent the instantiations of what was proven in earlier steps. Thus, in step 3 in the example, the expression is

$$\mathit{lube\text{-}oil}(O) \wedge \mathit{alarm}(A) \wedge \mathit{nn}(A, O) \supset \\ (\exists\, k, a_1, o_1) \mathit{physical\text{-}object}(k) \wedge \mathit{rel}(k, a_1) \\ \wedge\, \mathit{alarm}(a_1) \wedge \mathit{nn}(a_1, o_1) \wedge \mathit{lube\text{-}oil}(o_1)$$

Those prior instantiations consistent with higher constraints will be proven immediately from the antecedent, and new proofs will need to be discovered only for those which are inconsistent.[7]

3.3 The Algorithm for Constructing the Interpretation Expression

The required expression can be constructed by a recursive procedure which for convenience we will call *PRAG*. *PRAG* is called with a proposition and a logical expression as its two arguments. Initially, *PRAG* is called with the logical form of the sentence as its first argument and T as its second. The second argument (call it *expr*) will be used to build up the interpretation expression for the sentence.

[7] This technique is due to Mark Stickel.

First, to handle the congruence requirement imposed by the predicate p of the proposition on its arguments, if the knowledge base contains the selectional constraint

$$p(x) : r(x)$$

i.e., that r must be true of x, then $r(k) \wedge rel(k, a)$ is conjoined to *expr* where k is a new existentially quantified variable, and the relevant part of the logical form is altered from $p(a)$ to $p(k) \wedge rel(k, a)$

Next, each of the arguments is processed in turn. To resolve reference for an argument of the form $[a \mid P]$, all of the complex terms in P are replaced by their lead variables and the result is conjoined to *expr*.

Finally, for each of the arguments of the proposition, *PRAG* is called recursively on all of the conjuncts in its restriction P (with the original complex terms in P intact), and the results are conjoined to *expr*. *PRAG* returns the interpretation expression *expr*.

3.4 Minimality

Axioms can be assigned a cost, depending upon their salience. High salience, low cost axioms would then be tried first. Short proofs are naturally tried before long proofs. Thus, a cost depending on salience and length is associated wtih each proof, and hence with each interpretation. Where, as usually happens, there is more than one possible interpretation, the better interpretations are supported by less expensive proofs.

The second criterion for good interpretations is that we should favor the minimal solution in the sense that the fewest new entities and relations needed to be hypothesized. For example, the argument-relation pattern in nominal compounds, as in "lube oil pressure", is minimal in that no new implicit relation need be hypothesized; the one already given by the head noun will do. In metonymy, the identity coercion is favored for the same reason, and shorter coercions are favored over longer ones. Similarly, in the definite reference example (2), the air inlet valve of the mentioned compressor is favored over the air inlet valve of the compressor adjacent to the mentioned compressor, because of the same minimality prinicple.

These ideas at least give us a start on the very difficult problem of choosing the best interpretation.

4 Implicatures and Abduction

4.1 Given and New, Definite and Indefinite, Presupposed and Asserted

When we hear a sentence, we try to match part of the information it conveys with what we already know; the rest is new information we add (or decide not to add) to what we know. In our approach to reference, proving constructively from the knowledge base the existence of a definite entity is precisely the operation of matching the definite noun phrase with what we already know. Indefinite noun phrases, by contrast, require us to introduce a new entity, rather than find an already existing entity. However, a problem arises in the casreps that is really just an aggravated form of a problem that arises generally. There are virtually no articles. Sentence (1) was really

Disengaged compressor after lube oil alarm.

Consequently, we can almost never know whether an entity is definite or not. It can go either way. In

(6) Metal particles in oil sample and filter.

the oil filter is something we know about already. It is in our model of the device. "Oil filter" is definite. On the other hand, we are just being told that a sample of the oil was taken. "Oil sample" is indefinite.

In general discourse, where articles do occur, a problem still arises, since definite articles are sometimes used where the entity is not really known. If a speaker begins a sentence with

The trouble with John is ...

it may be that both the speaker and hearer know John has trouble and are able to resolve the reference. Or it could be that the speaker is introducing for the first time the fact that there is a problem with John. Related examples and an account of this phenomenon can be found in Hobbs (1987).

At first glance, it may seem that this problem is compounded in our ontologically promiscuous approach to logical form. There are entities corresponding to every predication made by the sentence, for example, the disengaging in sentence (1). For each of these entities we must decide whether it is definite or indefinite, and we are never given an article to tell us which it is. However, this turns out to be identical with the traditional problem of determining whether a predication is given or new, or in other terminology,

is part of the presuppositions of the sentence or part of what is asserted. Thus, the ontologically promiscuous notation, rather than compounding the definite-indefinite problem, collapses it and the given-new problem under a single treatment.

Normatively, the main verb of a sentence asserts new information and grammatically subordinated material is given. But this is not always true. In

> The philosophical Greeks contributed much to civilization.

it is unclear whether "philosophical" is intended to be used referentially as given information (the restrictive case) or is another new assertion being slipped into the sentence (the nonrestrictive case). In

> An innocent man was hanged today.

it could be that the speaker and hearer both know a man was hanged today, and the speaker is asserting his innocence. Where there is an adverbial, as in

> John saw his brother recently.

it is unclear (without intonation) whether the seeing or the recency or both is being asserted as new information.

A heuristic we tried initially was to assume that everything represented by an event variable ($e_1, e_2, \ldots$) corresponds to new information, i.e., is being asserted, and everything else is definite and is being used referentially. This is reasonably accurate in the casreps, but sentence (6) shows that it is not adequate everywhere. Consider also the text

> The low lube oil alarm sounded.
> The alarm was activated during routine start of start air compressor.

One can argue that the existence of an activation is already implicit in the sounding, and that therefore the activation is given, or definite.

The real story is that it is part of the job of pragmatics to determine whether each proposition in the sentence is being asserted or presupposed, and whether each noun phrase, regardless of surface form, is really definite or indefinite. This can be accomplished by means of referential implicatures, which is our current method for handling this problem.

4.2 Referential Implicatures

Let us begin with the simplest case—clear indefinites, as in

> A blade of the fan was chipped.

We cannot, at the outset, simply assert the existence of a B such that B is the blade of the fan, for we have not yet identified the fan. If we followed the naive search order of Section 3.2, we could wait until the fan was identified, assert the existence of one of its blades, and proceed to interpret the rest of the sentence. However, in the sophisticated search order, we cannot do this, for metonymy problems higher up in a logical form, say, for "chip", may need to be solved before reference problems lower down can be solved, and these metonymy problems will need information about its argument. Moreover, several fans may be proposed as the referent of "the fan", and B cannot be a blade of all of them. It must be the blade of the fan finally decided upon.

To handle this problem, as we process the sentence in the routine *PRAG*, we temporarily add to the knowledge base, statements asserting the existence of the indefinite entities. For indefinites at the bottom of the logical form, this is straightforward. For

> A metal chip was found in the sump.

we simply assert

$$(\exists\, y) metal(y) \wedge chip(y)$$

For indefinites that are functionally dependent on definites, things are a little more complicated. We cannot say

$$(\exists\, x, y) blade(x, y)$$

for there would be no guarantee the fan finally selected would be that y. We cannot say

$$(\forall\, y)(\exists\, x) blade(x, y)$$

for certainly not everything has a blade. We must make an assertion of the form

$$(\forall\, y) fan(y) \supset (\exists\, x) blade(x, y)$$

Think of this as saying, for any way that you can resolve "the fan", there is something which is its blade. But even this is not enough. It may be that we know about some fans that have no blades, and adding this assertion would make our knowledge base inconsistent. Thus, we need something more like the nonmonotonic assertion

$$(7) \quad (\forall y) fan(y) \wedge CONSISTENT[(\exists x) blade(x,y)] \supset (\exists x) blade(x,y)$$

In principle, this is what we believe is correct. The procedure *CONSISTENT* could be implemented by a procedural call within the theorem prover to the theorem prover itself. But of course, there is no guarantee it will terminate. So in practice, our present strategy is simply to assume consistency, ignoring the problem. A more principled approach would be to do some simple type-checking for inconsistencies, and if none are found, simply to assume consistency.

We may call assertions like (7) "referential implicatures".

Now let us return to the problem of Section 4.1, that it is impossible in general to know when a reference is definite or indefinite, or whether a proposition is presupposed or asserted. We can solve this problem by constructing referential implicatures for every entity in the logical form, whether from a definite, indefinite, or bare noun phrase, or a nonnominal reference. Of course, if this were all we did, every sentence would be easy to interpret and the interpretation would fail to tell us anything. For definite references, especially, we do not want to use the referential implicatures unless all else fails. To accomplish this, we associate costs with the various referential implicatures. Referential implicatures for explicitly indefinite NPs are free. The ones for explicitly definite NPs are quite expensive. Those for bare NPs are intermediate between the two, and those for events, introduced, for example, by verb phrases, are less expensive than those for bare NPs but not free. These costs are factored into the cost of proofs leading to interpretations, so that interpretations not making use of expensive referential implicatures are cheaper and hence better, if they are available. Thus, something is taken as new information only when it fails, after an appropriate amount of processing, to be recognized as given.

4.3 Identity Implicatures

A second kind of implicature that would be necessary in this kind of approach is an assumption, for no other reason than that it will lead to a

good interpretation of the text, that two entities are identical. The use of such implicatures for resolving pronoun references was discussed in Hobbs (1979). Here we will restrict our attention to their use in resolving nominal compounds.

Let us consider "oil sample" again. Suppose we have already inferred the existence of the oil—$oil(x)$. Suppose also we have assumed by the referential implicature the existence of a sample y of something z—$sample(y, z)$. We need to prove $nn(x, y)$. Axiom (3) tells us that if y is a sample of x then there is an nn relation between them. The only thing required for a proof is therefore an assumption that the oil y and the implicit second argument z of *sample* are identical. Since this would lead to a good interpretation, we are tempted to do this. However, we would like to check for consistency first. When we do some simple type checking, we find that z, since it can have a sample taken of it, must be a material, and we also find that the oil x is a material. This does not prove consistency, but it provides a coincidence of properties that at least makes an inconsistency less likely. So we go ahead and make the identification. A problem with this approach is that it is not clear how the drawing of identity implicatures can be triggered or controlled.

Grice (1975) gave the name "conversational implicature" to an assumption one had to make simply in order to get a good interpretation of a sentence. Referential implicatures and identity implicatures are particularly elementary and widespread cases of such assumptions.

4.4 Abduction and Redundancy

We are currently exploring a different approach to this whole family of problems—abductive reasoning. Pople (1973) and Cox and Pietrzykowski (1986) have proposed abductive reasoning as a means for diagnosis in expert systems. Abductive reasoning is reasoning to the best explanation. If we know $q(a)$ and we know $(\forall x)p(x) \supset q(x)$, then abductive reasoning leads us to conclude $p(a)$. Intuitively, $p(a)$ is our best guess for why the observed $q(a)$ is true. The problem with this is choosing the best $p(a)$ among a conceivably large set of possibilities. Both Pople (1973) and Cox and Pietrzykowski (1986) proposed choosing the *most* specific unprovable atom as the best explanation. Thus, an abscess in the liver is a better explanation than a pain in the chest. Stickel (1987) points out problems with this and argues that often in natural language interpretation, the least specific unprovable atom is the most appropriate one to be assumed. Thus, if "a fluid" is mentioned, we should not assume it is lube oil.

A generalization of this kind of abductive capability is now being implemented in the KADS theorem prover. It will allow us to recast the whole problem of definite and indefinite reference. The interpretation expression will be constructed as before. Instead of referential implicatures being asserted with their associated costs, the same costs would now be attached to the atoms to be proved as the cost of simply assuming them. The atoms will be assumed with their most specific bindings, which will perform the function of including the antecedents in the referential implicatures. Therefore, if a definite reference is resolvable with respect to the knowledge base, it will be resolved with a proof considerably cheaper than one requiring the assumption of the existence of an entity of that description. However, if it is not resolvable, its existence will be assumed.

This approach also gives us a way of dealing with examples like

> Investigation revealed adequate lube oil saturated with metal particles.

Here, "lube oil" is given information, while "adequate" and "saturated with metal particles" are new. Under the abductive approach *lube-oil*(x) will be resolved with the corresponding atom in the domain model, the binding will propagate to *adequate*(x) and *saturate*(ps, x), and these instantiated atoms will then be assumed. Solving this problem using referential implicatures would be extremely cumbersome.

There is a further possible benefit from the abductive approach; it may take the place of identity implicatures and allow us at last to exploit the natural redundancy of all discourse. An example can illustrate this best. Consider the sentence

> Inspection of lube oil filter revealed metal particles.

There are several coreference problems involving implicit arguments. We would like to be able to discover that the person doing the inspection was the same as the person to whom the particles were revealed, and we would like to know that the metal particles were found in the lube oil filter. This information is not explicit in the sentence. The general problem is to discover the coreference relations among arguments in syntactically independent regions of a sentence.

Let us unpack the words in the sentence to see the overlap of semantic content. If x inspects y, then x looks at y in order that this looking will cause x to learn some property relevant to the function of y. In order to avoid quantifying over predicates, let us assume an analysis of location, or

at, that allows properties metaphorically to be located at entities. Then we can state formally,

$$\begin{aligned}&(\forall e_1, x, y) inspect'(e_1, x, y) \equiv \\ &\quad (\exists e_2, e_3, z, e_4) look\text{-}at'(e_1, x, y) \wedge cause(e_1, e_2) \\ &\qquad \wedge learn'(e_2, x, e_3) \wedge at'(e_3, z, y) \wedge relevant\text{-}to(e_3, e_4) \\ &\qquad \wedge function(e_4, y)\end{aligned}$$

If an event e_1 reveals z to x, then there is a y such that e_1 causes x to learn that z is at y. Formally,

$$\begin{aligned}&(\forall e_1, z, x) reveal(e_1, z, x) \equiv \\ &\quad (\exists e_2, e_3, y) cause(e_1, e_2) \wedge learn'(e_2, x, e_3) \wedge at'(e_3, z, y)\end{aligned}$$

A filter is something whose function is to remove particles. Formally,

$$\begin{aligned}&(\forall e_6, y, w) filter'(e_6, y, w) \equiv \\ &\quad (\exists e_4, z, s) function(e_4, y) \wedge remove'(e_4, y, z, w) \wedge particle(z) \\ &\qquad \wedge typical\text{-}element(z, s)\end{aligned}$$

If y removes z from w, then there is a change from z's being in w to z's being at y.

$$\begin{aligned}&(\forall e_4, y, z, w) remove'(e_4, y, z, w) \equiv \\ &\quad (\exists e_8, e_3) change'(e_4, e_8, e_3) \wedge in'(e_8, z, w) \wedge at'(e_3, z, y)\end{aligned}$$

Finally, let us say the end point of a change is relevant to the change.

$$(\forall e_4, e_8, e_3) change'(e_4, e_8, e_3) \supset relevant\text{-}to(e_3, e_4)$$

Now the interpretation expression will include

$$\begin{aligned}&inspect'(e_1, x_1, y) \wedge reveal(e_1, z, x_2) \wedge filter'(e_6, y, w) \wedge particle(z) \\ &\quad \wedge typical\text{-}element(z, s)\end{aligned}$$

If the above axioms are used to expand this expression, then the operation that Stickel calls "factoring" and Cox and Pietrzykowski call "synthesis" can apply; we can unify goal atoms wherever possible. We can thus unify the variables as indicated in the way we have named them in the axioms. Further suppose that atoms resulting from factoring have enhanced assumability, since they will lead to minimal interpretations. If we assume those atoms, then we will have concluded that the inspector x_1 and the beneficiary x_2 of the revealing are identical and that the particles are in the filter.

One difficulty with is approach is the possible inefficiency introduced by allowing the results of factoring to be assumable. Another difficulty is whether the bidirectional implications in the above axioms are really justified, and how the procedure could be made to work if we only had implication to the right. These issues are under investigation.

5 Implementation

In our implementation of the TACITUS system, we are beginning with the minimal approach and building up slowly. As we implement the local pragmatics operations, we are using a knowledge base containing only the axioms that are needed for the test examples. Thus, it grows slowly as we try out more and more texts. As we gain greater confidence in the pragmatics operations, we move more and more of the axioms from our commonsense and domain knowledge bases into the system's knowledge base. Our initial versions of the pragmatics operations are, for the most part, fairly standard techniques recast into our abstract framework. When the knowledge base has reached a significant size, we will begin experimenting with more general solutions and with various constraints on those general solutions.

To see what the program does, let us examine its output for one sentence.

```
Tacitus> operator was unable to maintain lo pressure to sac
```

"Lo" is an abbreviation for "lube oil" and "sac" is an abbreviation for "starting air compressor". The sentence is parsed and six parses are found. Prepositional phrase attachment ambiguities are merged to reduce the number of readings to four. The highest ranking parse is the correct one because the adjective complement interpretation is favored over the purpose clause interpretation for infinitive clauses, and because the attachment of "to sac" to "pressure" is favored both by a heuristic that favors right attachment and one that favors argument prepositions attached to their relational nouns. The logical form is produced for this parse. It can be read "In the past there was a condition E12 which is the condition of X1 being unable to do E3 where E3 is the possible event of X1, who is the operator, maintaining X4, which is the pressure of something Y1 at X10, which is the starting air compressor (and, by the way, is not identical to X4), and there is some implicit relation NN between X6, which is lube oil, and X4.

```
OPERATOR PAST1 BE UNABLE TO MAINTAIN LO PRESSURE TO SAC
six parses were found

After merging ambiguities, there are four logical forms
The Highest Ranking LF:
(E (E13 E12 E2 X4 E11 X10 Y1 E5 E7 X6 E8 E3 X1)
   (PAST! E13
          (E12 (UNABLE! E12 X1
                   (E3 (MAINTAIN! E3
                           (X1 (OPERATOR! E2 X1))
                           (X4 (PRESSURE! E5 X4 Y1
                                   (X10 (SAC! E11 X10)
                                        (NOT= X10 (X4))))
                               (NN! E8 (X6 (LUBE-OIL! E7 X6))
                                    X4))))))))
```

The sentence is interpreted from the inside out, so the first problem is finding the reference of "operator". "BARE" means there is no determiner.

```
Reference Problem: X1: treated as type BARE
I|
Prove: (E (x1 e2)
          (Operator! e2 x1))

I|.V
```

The reference is resolved by unifying `x1` with the constant `opr1` in the axioms that encode the domain model. `opr1` has the property `Operator`.

```
Reference Resolved:
x1 = opr1
```

This was established by inferring the following proposition from the axioms. `operator-ness1` is the condition of `opr1`'s having the property `Operator`.

```
Inferred the following propositions:
           (Operator! operator-ness1 opr1)
```

The next problem is the reference of "sac". We do not use the noncoreference information encoded by Not= at the present time. It is always assumed to be true. The reference is resolved by identifying the sac as the one mentioned in the domain model.

```
Reference Problem: X10: treated as type BARE
I|I|
Prove: (E (x10 e11 x4)
          (AND (Not= x10 cons(x4,nil))
               (Sac! e11 x10)))

ID*|.VV

Reference Resolved:
x10 = sac1

Inferred the following propositions:
           (Not= sac1 cons(X195,nil))
           (Sac! sac-ness1 sac1)
```

The next problem, moving from the inside out, is to satisfy the constraints the word "pressure" places on its arguments. A coercion constant k3, which is related to the entity sac1 that we have already resolved X10 to, is introduced to take care of the possibility of metonymy. The word "pressure" requires that y1 must be a fluid that can be located at k3.

```
Metonymy Problem:
(PRESSURE! E5 X4 Y1 X10)
III|||I|
Prove: (E (k3 y1 k5 k4 x4)
          (AND (Not= sac1 cons(x4,nil))
               (Fluid! k4 y1)
               (At! k5 y1 k3)
```

```
            (Related k3 sac1)))
```

The stars and bars tell the user that the theorem prover is working away.

```
ID*|***|*|*|***|*|.T.*
```

One way of being related is being a part of, and the bearings are a part of the sac, and the only fluid that the system currently knows about that can be at something related to the sac is the lube oil. So it is determined that it must be the pressure of the lube oil at the bearings, which are a part of the sac. Had the system also known about air, it could have come up with a different interpretation. This is an example where the compound nominal, and thus the reference, problem for "pressure" should have been done at the same time, and where exploiting the redundancy of information encoded in the words "lube oil" and "pressure" would have helped.

The instantiated inference steps are listed. Lube oil is known to be a fluid because oil is and lube oil is oil. It is known to be at the bearings because it is known that the pump transmits lube oil from the pump to the bearings, and the being located is the end state of that transmission. The bearings are a part of the sac because they are a part of the lube oil system, which is a part of the sac.

```
Metonymy Resolved:
y1 = lube-oil1
x10 = sac1
k3 = bearings1

Inferred the following propositions:
           (Partof bearings1 sac1)
           (Not= sac1 cons(X206,nil))
           (Fluid! k4 lube-oil1)
           (Oil! oil-ness-11(_) lube-oil1)
           (Lube-Oil! lube-oil-ness1 lube-oil1)
           (At! k5 lube-oil1 bearings1)
           (Transmit! transmit-ness2 pump1 lube-oil1 pump1
                      bearings1)
           (Related bearings1 sac1)
```

```
            (Component! component-ness1 losys1 sac1)
            (Component! component-ness3 bearings1 losys1)
            (Partof losys1 sac1)
```

The fact that there has been a coercion is reported to the user.

```
Coercion: (Pressure! e5 x4 y1 k3)
```

Next is the reference problem for "lube oil", which is solved in the same way as the two previous reference problems.

```
Reference Problem: X6: treated as type BARE
I|*|I|*|
Prove: (E (x6 e7)
          (Lube-Oil! e7 x6))

I|.VV

Reference Resolved:
x6 = lube-oil1

Inferred the following propositions:
            (Lube-Oil! lube-oil-ness1 lube-oil1)
```

The reference problem for "pressure" is addressed with its arguments instantiated with the values that have already been discovered. If this were inconsistent, the system would back up, and try to prove the fail-soft version of the interpretation expression described in Section 3.2. The compound nominal interpretation problem is dealt with here as well. It is solved because the relational noun - argument relation is one possible way for Nn to be true.

```
Reference Problem: X4: treated as type BARE
I|I|I|*|I|
Prove: (E (x4 e5 e8)
```

```
          (AND (Nn! e8 lube-oil1 x4)
               (Pressure! e5 x4 lube-oil1 bearings1)))

I|***|**********|||.|||*||

Reference Resolved:
x4 = pressure1
x6 = lube-oil1
k3 = bearings1
y1 = lube-oil1

Inferred the following propositions:
           (Nn! e8 lube-oil1 pressure1)
           (Pressure! pressure-ness1 pressure1 lube-oil1
                      bearings1)
```

The metonymy problem for the predicate MAINTAIN is handled next. For something to be maintained, it must be an eventuality that is desired by the maintainer. The adequacy of the lube oil pressure, being a normal condition, is desired by the operator. Hence, "maintain lube oil pressure" is coerced into "maintain the adequacy of lube oil pressure".

```
Metonymy Problem: (MAINTAIN! E3 X1 X4)
IIIID|||ID*|
Prove: (E (k10 k11 k12)
          (AND (Eventuality k11)
               (Desire! k12 k10 k11)
               (Related k11 pressure1)
               (Related k10 opr1)))

ID*|***|*|.T.*

Metonymy Resolved:
x4 = pressure1
k11 = adequate-ness1
x1 = opr1
k10 = opr1
```

```
Inferred the following propositions:
           (Pressure! pressure-ness1 pressure1 lube-oil1
                      bearings1)
           (Adequate! adequate-ness1 pressure1)
           (Related opr1 opr1)
           (Desire! k12 opr1 adequate-ness1)
           (Normal adequate-ness1)
           (Related adequate-ness1 pressure1)

Coercion: (Maintain! e3 opr1 k11)
```

The system also tries to solve nonnominal reference problems. Here it seeks to determine if it already knows about a maintaining event. It does not, so a referential implicature introduces it as a new entity.

```
Reference Problem: E3: treated as type EVENT
I|*|ID*|
Prove: (E (e3)
          (Maintain! e3 opr1 adequate-ness1))

I|.*

New Entity Introduced:
         E3
```

The constraint UNABLE places on its arguments is that E3 must be an eventuality. This is verified. A possible coercion is assumed by introducing the coercion constant k15, but identity is one way of being coerced.

```
Metonymy Problem: (UNABLE! E12 X1 E3)
IID|ID*|
Prove: (E (k15)
          (AND (Eventuality k15)
               (Related k15 maintain-ness-72)))

ID*|.*.**
```

```
Metonymy Resolved:
e3 = maintain-ness-72
k15 = maintain-ness-72

Inferred the following propositions:
           (Related e3 e3)
```

Nonnominal reference is determined for the inability as well, and it is determined to be new.

```
Reference Problem: E12: treated as type EVENT
I|*|ID*|
Prove: (E (e12)
          (Unable! e12 opr1 maintain-ness-72))

I|.*

New Entity Introduced:
 E12

I=|*|
```

This completes the interpretation of the sentence. All of the properties that have been inferred are listed. Those properties that required referential implicatures are new information and are listed as such.

```
INTERPRETATION OF SENTENCE:


 New Information:

e13:                 (Past! e13 e12)
e12:                 (Unable! e12 opr1 e3)
e3:                  (Maintain! e3 opr1 adequate-ness1)


 Old Information:
```

```
opr1:               (Operator! operator-ness1 opr1)
                    (Desire! k12 opr1 adequate-ness1)
adequate-ness1:     (Adequate! adequate-ness1 pressure1)
                    (Normal adequate-ness1)
                    (Related adequate-ness1 pressure1)
pressure1:          (Pressure! pressure-ness1 pressure1
                               lube-oil1 bearings1)
                    (Nn! e8 lube-oil1 pressure1)
lube-oil1:          (Fluid! k4 lube-oil1)
                    (Oil! oil-ness-11 lube-oil1)
                    (Lube-Oil! lube-oil-ness1 lube-oil1)
                    (At! k5 lube-oil1 bearings1)
                    (Transmit! transmit-ness2 pump1 lube-oil1
                               pump1 bearings1)
bearings1:          (Component! component-ness3 bearings1
                                losys1)
                    (Related bearings1 sac1)
                    (Partof bearings1 sac1)
losys1:             (Partof losys1 sac1)
                    (Component! component-ness1 losys1 sac1)
sac1:               (Sac! sac-ness1 sac1)
pump1:


I=I=I=I=I=I=I=I=I=I=I=I=DDD||||||||||||
```

The interpretation of the sentence makes no assumptions about the existential status of the various eventualities conveyed by the sentence. This is done in a final phase of processing. The highest level eventuality is assumed to exist, and decisions are propagated down from there. Thus, since the past-ness exists, the inability exists. Since the inability exists, the maintaining does not exist. Since it does not exist, neither does the adequacy. That is all that can be concluded for sure. Simply as a heuristic, the other eventualities are assumed to exist.

```
Assuming the following eventualities do exist:
          E12, E13, E8, K12, K4, K5, LUBE-OIL-NESS1,
          OPERATOR-NESS1, PRESSURE-NESS1, SAC-NESS1
```

```
Assuming the following eventualities do not exist:
        ADEQUATE-NESS1, E3
```

Acknowledgements

The authors have profited from discussions with Mark Stickel, Doug Edwards, Mabry Tyson, Bill Croft, Fernando Pereira, Ray Perrault, and Stu Shieber about this work. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

References

[1] Cox, P. T., and T. Pietrzykowski, 1986. "Causes for Events: Their Computation and Applications", Proceedings, CADE-8, pp. 608-621.

[2] Downing, Pamela, 1977. "On the Creation and Use of English Compound Nouns", *Language*, vol. 53, no. 4, pp. 810-842.

[3] Finin, Timothy, 1980. "The Semantic Interpretation of Nominal Compounds", Report T-96, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, June 1980.

[4] Grice, H. P., 1975. "Logic and Conversation", in P. Cole and J. Morgan, eds., *Syntax and Semantics*, vol. 3, pp. 41-58, Academic Press, New York.

[5] Grosz, Barbara, Norman Haas, Gary Hendrix, Jerry Hobbs, Paul Martin, Robert Moore, Jane Robinson, Stanley Rosenschein, 1982. "DIALOGIC: A Core Natural-Language Processing System", Technical Note 270, Artificial Intelligence Center, SRI International.

[6] Grosz, Barbara J., Douglas E. Appelt, Paul Martin, Fernando C. N. Pereira and Lorna Shinkle, 1985. "The TEAM Natural-Language Interface System", Final Report, Project 4865, Artificial Intelligence Center, SRI International, Menlo Park, California.

[7] Hobbs, Jerry R., 1979. "Coherence and Coreference", *Cognitive Science*, vol. 3, no. 1, pp. 67-90.

[8] Hobbs, Jerry R., 1983. "An Improper Treatment of Quantification in Ordinary English", *Proceedings, 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts, pp. 57-63.

[9] Hobbs, Jerry R., 1985. "Ontological Promiscuity", *Proceedings, 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, Illinois, pp. 61-69.

[10] Hobbs, Jerry R., 1985. "Implicature and Definite Reference", Report No. CSLI-87-99, Center for the Study of Language and Information, Stanford University, Stanford, California, May 1987.

[11] Hobbs, Jerry R., William Croft, Todd Davies, Douglas Edwards, and Kenneth Laws, 1986. "Commonsense Metaphysics and Lexical Semantics", *Proceedings, 24th Annual Meeting of the Association for Computational Linguistics*, New York, June 1986., pp. 231-240.

[12] Levi, Judith, 1978. *The Syntax and Semantics of Complex Nominals*, Academic Press, New York.

[13] Moore, Robert C., 1981. "Problems in Logical Form", *Proceedings, 19th Annual Meeting of the Association for Computational Linguistics*, Stanford, California, pp. 117-124.

[14] Nunberg, Geoffery, 1978. "The Pragmatics of Reference", Ph. D. thesis, City University of New York, New York.

[15] Pople, Harry E., 1973. "On the Mechanization of Abductive Logic", Proceedings, International Joint Conference on Artificial Intelligence, Stanford, California, August 1973, pp. 147-152.

[16] Stickel, Mark E., 1982. "A Nonclausal Connection-Graph Theorem-Proving Program", *Proceedings, AAAI-82 National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, pp. 229-233.

[17] Stickel, Mark E., 1987. "Pragmatics as Abduction: Least-Specific Abduction and its Use in Natural-Language Interpretation", manuscript.

[18] Woods, William, 1977. "Semantics and Quantification in Natural Language Question Answering", *Advances in Computers*, Volume 17, Academic Press, New York, pp. 1-87.